



Nunez-Yanez, J.L., & Chouliaras, V.A. (2005). High-performance arithmetic coding VLSI macro for the H264 video compression standard. *IEEE Transactions on Consumer Electronics*, 51(1), 144-151. <https://doi.org/10.1109/TCE.2005.1405712>

Early version, also known as pre-print

Link to published version (if available):
[10.1109/TCE.2005.1405712](https://doi.org/10.1109/TCE.2005.1405712)

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

High-performance Arithmetic Coding VLSI Macro for the H264 Video Compression Standard

J. L. Núñez, V. A. Chouliaras, Member, IEEE

Abstract — This paper investigates the algorithmic complexity of arithmetic coding in the new H264 video coding standard and proposes a coprocessor to reduce it by more than an order of magnitude. The coprocessor is based on an innovative algorithm named as the MZ-coder and maintains the original coding efficiency with a multiplication-free, non-stalling, fully pipelined architecture with modest hardware requirements. The coprocessor delivers a constant throughput for both coding and decoding of 1 bit per cycle and can be attached to a controlling CPU whose ISA has been extended with arithmetic coding instructions.

Index Terms — arithmetic coding, H264, video coding, Golomb codes, renormalization.

I. INTRODUCTION

The exponential increase in the amount of digital visual information that must be transmitted and stored efficiently has motivated a large body of research into advanced video coding techniques which allows orders of magnitude reduction in the required bit-rates. New video coding standards such as the recent H264 video codec (also known as MPEG4 part 10) [2] deliver better quality and lower bit rates but at the expense of an almost exponential increase in the number of CPU cycles required per input frame of video data. The introduction of advanced entropy coding within the H264 standard with the pioneering use of context-based arithmetic coding [3] in a lossy video standard is one of the reasons behind the increase in the computational cost of the codec. The high-speed arithmetic coder (AC) coprocessor described in this paper achieves a significant reduction of the AC computational cost in the H264 video codec with modest hardware complexity.

II. HARDWARE-BASED BINARY ARITHMETIC CODERS

The IBM Q-coder and the QM-coder [4] are the best known examples of hardware-based binary arithmetic coders. A VLSI implementation of both the Q-coder and QM-coder has been done in [5]. The device is called the Qx-coder and can implement both algorithms clocking at 75 MHz with a throughput of around 64 Mbits/second using 0.35 μ m standard cell technology from IBM (CMOS 5S). The adaptive binary

arithmetic coding chip presented in [6] replaces the division operation by storing the probability values in a look table and using the coder state as a pointer to a particular probability in that table. Multiplications on the other hand are done explicitly using 8x8 parallel multiplier.

III. ANALYSIS OF AC IN THE H264 VIDEO CODEC

The original arithmetic coding implementation in the H264 codec is known as CABAC [8]. Preliminary profiling of the H264 algorithm revealed the average number of calls to the AC routines per frame as a function of the quantization parameter QP as shown in Fig. 1.

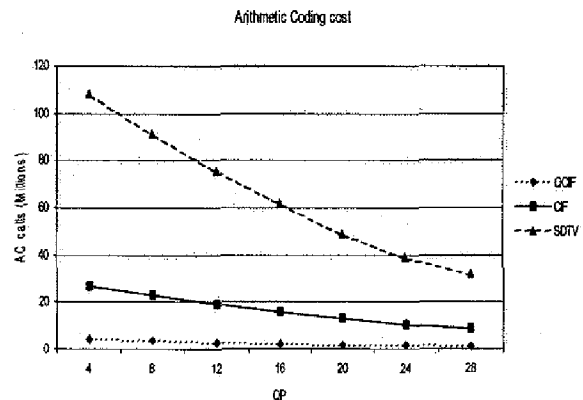


Figure 1. H264 arithmetic coding complexity

Fig. 1 shows that AC is a very compute intensive operation and since traditional parallelizing techniques such as SIMD extensions cannot accelerate this essentially sequential process, the introduction of dedicated hardware support in the form of a specialized coprocessor, is a suitable solution.

IV. PROPOSED ARITHMETIC CODING ALGORITHM

The MZ-coder evolves from the Z-coder software algorithm presented in [7] as a generalization of the well known Golomb/Rice coder for lossless coding of bilevel images. Fig. 2 shows the simplified (pseudocode) description of the MZ-coder algorithm (right) and the original CABAC algorithm (left). The coder state variables are range and low for the CABAC algorithm and range and subend for the MZ-coder.

The renormalization process in the MZ-coder does not include internal dependencies. As a result it can be readily accomplished with a single shift left operation. On the other hand the pseudocode for the CABAC algorithm shows the internal dependencies of low inside the while loop. This dependency means that a variable number of cycles (from 0

J. L. Núñez is with the Department of Electronic and Electrical Engineering, University of Bristol, UK (e-mail: j.l.nunez@bristol.ac.uk)

V. A. Chouliaras is with the department of Electronic Engineering, University of Loughborough, UK (e-mail: v.a.chouliaras@lboro.ac.uk)

up to a maximum of 6) are required to maintain the state variables in the required range.

```

rLPS = table256x8 (state,range);
range = range - rLPS;
if ( symbol != MPS )
{
    low += range;
    range = rLPS;
}
/*renormalization loop*/
while( range < QUARTER)
{
    if(low >= HALF)
    {
        Output bit;
        low -= HALF;
    }
    else
    {
        if (low < QUARTER)
            Output bit;
        else
            bits to follow++;
        low -= QUARTER;
    }
    low <= 1;
    range <= 1;
}

pLPS = table64x6(state);
Z = range + pLPS;
if (Z >= HALF)
    Z = QUARTER + Z >> 1;
if (symbol == MPS)
{
    range = Z;
    if (range >= HALF)
    {
        output 1 bit;
        range <= 1;
        subend <= 1;
    }
    Z = FULL - Z;
    subend += Z;
    range += Z;
    shift_bits = shift(range);
    output shift_bits bits;
    range <= shift_bits;
    subend <= shift_bits;
}

```

Figure 2. CABAC & MZ pseudocode description

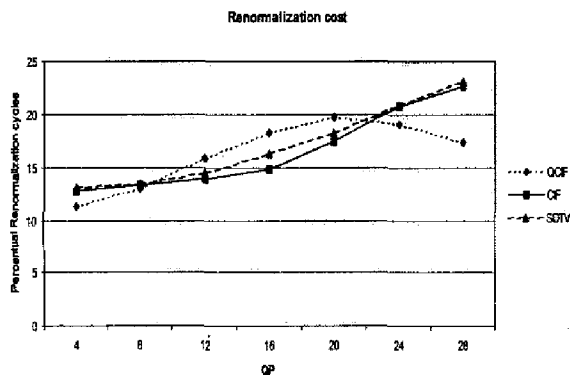


Figure 3. Renormalization costs in CABAC

Fig. 3 illustrates that the costs of multiple-cycle renormalization account for a throughput degradation of around 15% in the original CABAC algorithm.

V. ARCHITECTURE AND VLSI IMPLEMENTATION

Fig. 4 shows the hardware architecture of the arithmetic coding/decoding coprocessor. The chosen VLSI technology was the UMC 0.13 μm , 8-copper silicon process. The maximum operating frequency was 330 MHz worst-case (throughput of 330 Mbits/second) and the complexity of both the coder and decoder is 5600 standard cells.

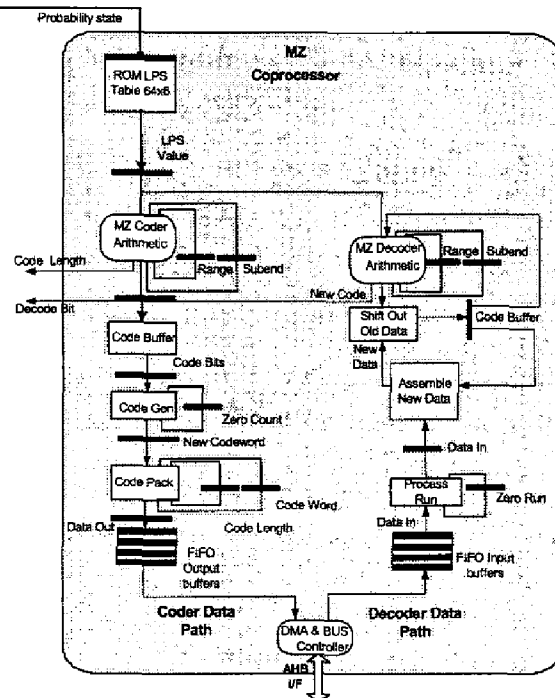


Figure 4. Coprocessor architecture

VI. CONCLUSIONS

The H264 video coding standard is expected to become the near-future enabling technology for personal multimedia communications. Major efforts are currently underway within industry and academia to accelerate the compute intensive motion estimation, transform and quantization functions through developing fast algorithms and exploiting the available data level parallelism. Entropy coding based on arithmetic coding is a mainly sequential process, not well suited to this kind of optimization. Its acceleration with the proposed hardware architecture can play a major role in bringing real-time H264 video coding within the grasp of low-power embedded devices.

References

- [1] G. Lawton, "New Technologies Place Video in Your Hand", IEEE Computer, Vol. 34, No. 4, pp. 14-17, 2001.
- [2] G. Bjontegaard, "H.26L Test Model Long Term Number 4 (TML 4) draft0", ITU-TSG16/Q.6 Q15-J-72, June 2000.
- [3] G. Langdon, "An Introduction to Arithmetic Coding", IBM J. Res. Develop, Vol. 28, No. 2, pp. 135-149, March 1984.
- [4] W.B. Pennebaker et al, "An overview of the Basic Principles of the Q-Coder Adaptive Binary Arithmetic Coder" IBM J. Res. Develop, Vol 32, No. 6, pp. 717-725, November 1988.
- [5] M. J. Slattery, J. L. Mitchell, "The Qx-coder", IBM Journal of Research and Development, Vol. 42, No. 6, pp. 767-784, 1998.
- [6] D. Marpe, H. Schwartz, T. Wiegand, "Context-Based Adaptive Arithmetic Coding in the H.264 Video Compression Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, pp. 620-636, 2003.
- [7] L. Bottou, P. G. Howard, Y. Bengio, "The Z-coder adaptive binary coder", In Proceedings of the Data Compression Conference, pages 13-22, March 1998.